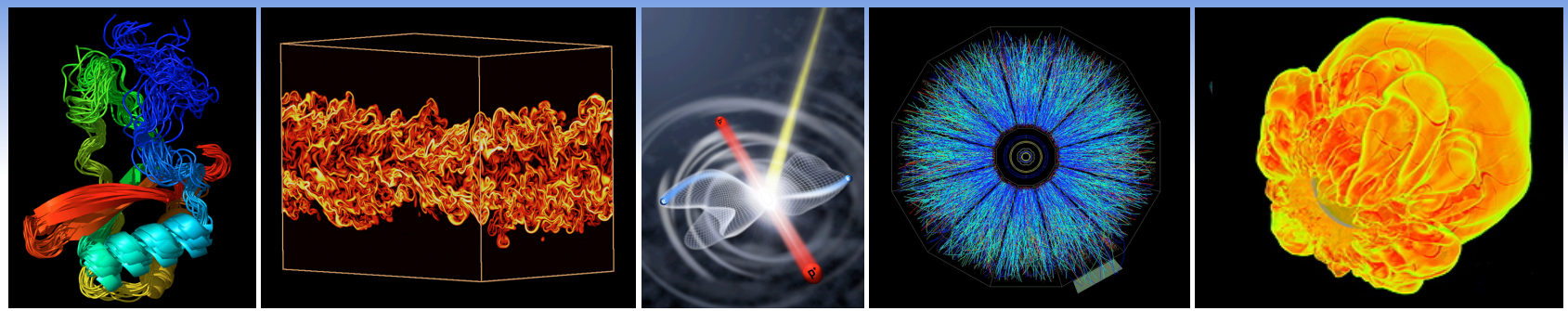


Hardware and Software in the Multicore Era

Katherine Yelick
NERSC

HEPiX Meeting, October 26, 2009

U.S. DEPARTMENT OF ENERGY | Office of Science

NERSC

BERKELEY LAB

NERSC
NATIONAL ENERGY RESEARCH SCIENTIFIC COMPUTING CENTER

Experience and Plans for High Energy Physics Computing at NERSC

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB

NERSC
NATIONAL ENERGY RESEARCH SCIENTIFIC COMPUTING CENTER

Cosmic Microwave Background

Objective: **Analyze data from the Planck satellite -- definitive Cosmic Microwave Background (CMB) data set.**

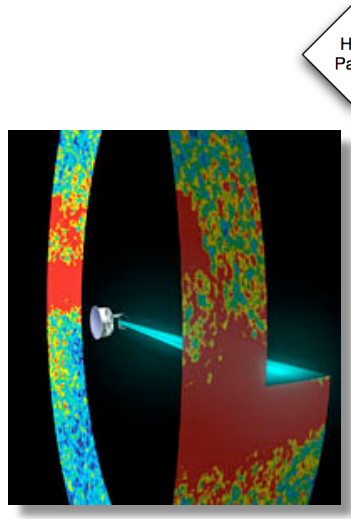
Implications: **CMB: image of the universe at 400k years, relic radiation from Big Bang. CMB Nobel prize in 2006**

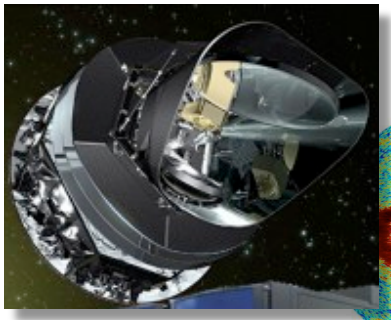
Accomplishments: **NERSC provides the components of the data pipeline for noise reduction, map-making, power spectrum analysis, and parameter estimation.**

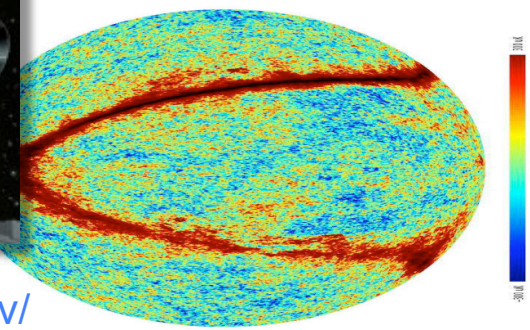
Data sets analyzed as a whole because complex data correlations; no "divide and conquer"

- **32 TB final data set size, ~400 users**
- **Launched May09, first "light" Sept09**
- **Also ~10k-core XT4 MonteCarlo calibration runs, produce ~10X data**
- **Anticipate Moore's law growth in data set size for *15 years***

PI: J. Borrill (LBNL)







https://c3.lbl.gov/

BERKELEY LAB

NERSC
NATIONAL ENERGY RESEARCH SCIENTIFIC COMPUTING CENTER

KamLAND

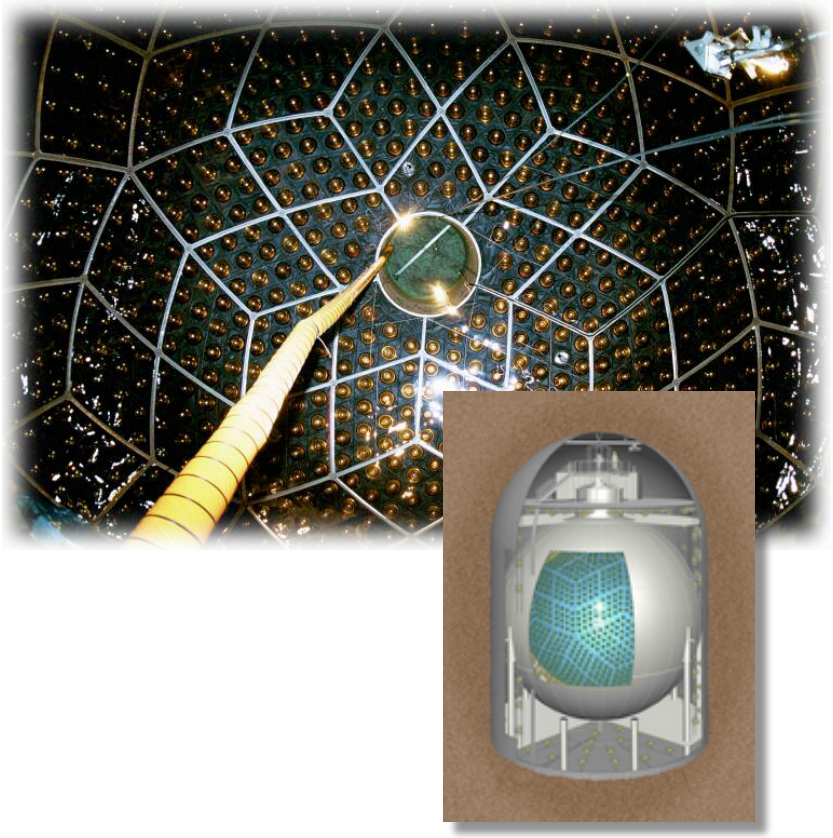
Objective: **Archive, analyze all stages of the US data from Kamioka Liquid Scintillator Anti-Neutrino Detector**

Implications: **Substantially increased our scientific knowledge of neutrinos**

Accomplishments: **Many significant physics milestones – neutrino oscillation, precise value for the neutrino oscillation parameter, etc.**

- **NERSC resources instrumental in reactor neutrino analysis and the preparations for the solar phase;**
- **Currently recording data at trigger rate of 100Hz, data rate of 200GB/day, 365 days/yr**
- **0.6 PB of data stored from 6 years; plan to read large fraction of this in 2010**

PI: S. Freedman (UCB)



U.S. DEPARTMENT OF ENERGY | Office of Science

http://kamland.lbl.gov/

BERKELEY LAB

ALICE

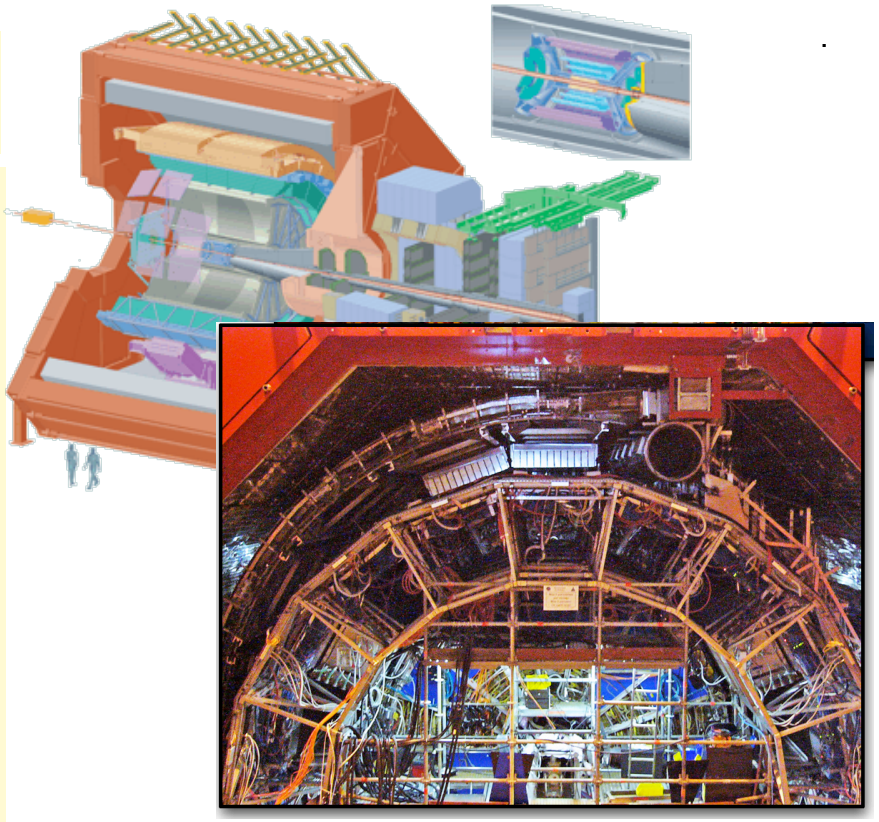
Objective: **Data analysis and simulations for the ALICE heavy-ion detector experiment at the LHC.**

Implications: **Understanding of dense QCD matter.**

Notes: **Uses (primarily) NERSC's PDSF cluster + LLNL + Grid resources;**

- **Expect ~600TB of data in 1GB files, ~25% of USA obligation in 2010.**
- **Challenge of providing direct-charged resources for experimentation that might be delayed.**
- **Simulations to reconstruct and analyze detector events prior to experiment.**
- **Longer term: Estimate 3.8 PB of disk space and 5.31 PB of HPSS in 2013, accessible by international community.**

PI: P. Jacobs (LBNL)



http://aliceinfo.cern.ch/Collaboration/

Palomar Transient Factory

***Objective*: Process, analyze & make available data from Palomar Transient Sky survey (~300 GB / night) to expose rare and fleeting cosmic events.**

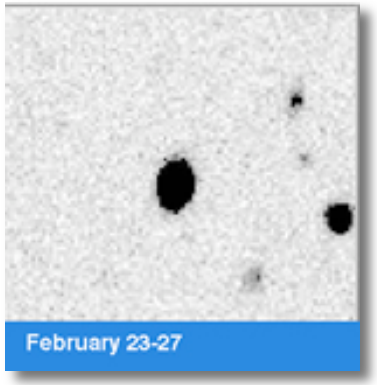
***Implications*: First survey dedicated solely to finding transient events.**

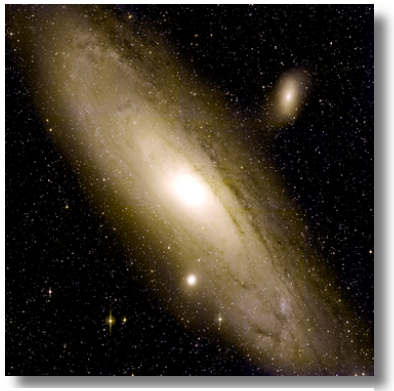
***Accomplishments*: Automated software for astrometric & photometric analysis and *real-time* classification of transients.**

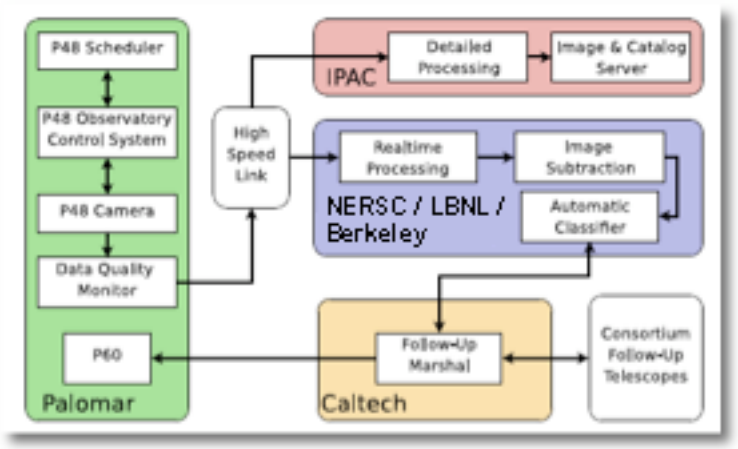
- **Analysis at NERSC is fast enough to reveal transients *as data are collected.***
- **Has *already uncovered* more than 40 supernovae explosions since Dec., 2008.**
- **Uncovering a new event about *every 12 minutes.***

- **40k hours, 1M Storage units for tape in 2009; Uses NERSC's 400-TB NGF + gateway**

PI: P. Nugent (LBNL)







PTF project data flow

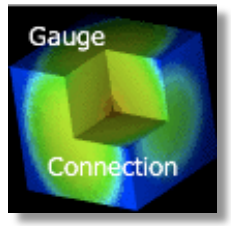
Two manuscripts submitted to Publications of the Astronomical Society of the Pacific

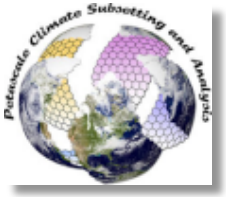
Observations

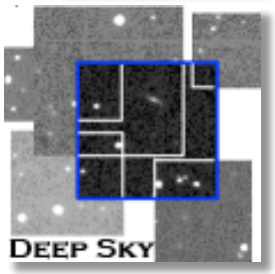
- Few projects are purely simulation or observation.
- It's not just about providing computers / disks / tapes / fiber
  - It's about organization and secure, public access with modern tools
- Some data sets becomes too large to move "home."
- Value of data varies:
  - Observations may be irreplaceable
  - Simulation data becomes less important over time
- Fast I/O is key
- Manipulation and analysis of data is a growing problem
  - More computing capacity will be needed to handle the data
  - Some data sets can only be addressed only by large HPC systems

Science Gateways

- **Create scientific communities around data sets**
  - NERSC HPSS, NGF accessible by broad community for exploration, scientific discovery, and validation of results
  - Increase value of existing data
- ***Science gateway: custom* hardware, software to provide remotely data/computing services**
  - Deep Sky – "Google-Maps" for astronomical image data
    - Discovered 36 supernovae in 6 nights during the PTF Survey
    - 15 collaborators worldwide worked for 24 hours non-stop
  - GCRM – Interactive subselection of climate data (pilot)
  - Gauge Connection – Access QCD Lattice data sets
  - Planck Portal – Access to Planck Data
- **New models of computational access**
  - Projects with mission-critical time constraints require guaranteed turn-around time.
  - Reservations for anticipated needs: Computational Beamlines
  - Friendly interfaces for applications and workflows







Deep Sky Science Gateway

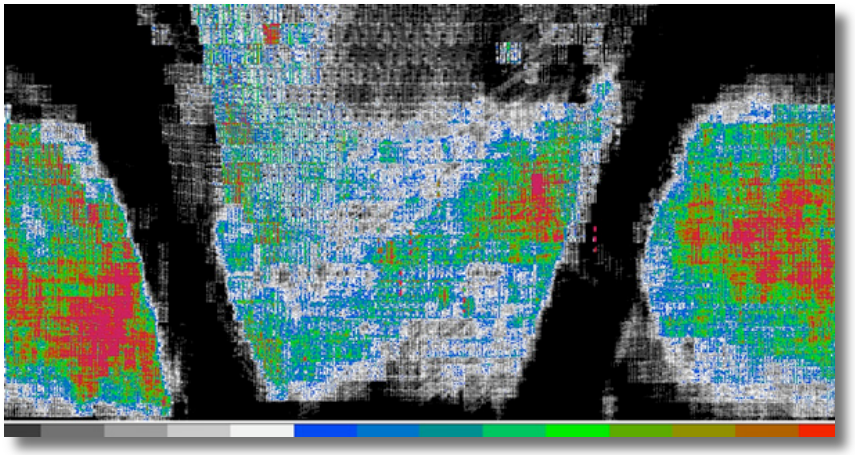
Objective: **Pilot project to create a richer set of compute- and data-resource interfaces for next-generation astrophysics image data, making it easier for scientists to use NERSC and creating world-wide collaborative opportunities.**

Implications: **Efficient, streamlined access to massive amounts of data – some archival, some new -- for broad user communities.**

Accomplishments: **Open-source Postgres DBMS customized to create Deep Sky DB and interface: www.deepskyproject.org**

- **90TB of 6-MB images stored in HPSS / NGF (biggest NGF project now)**
  - **-- images + calibr. data, ref. images, more**
  - **-- special storage pool focused on capacity not bandwidth**
- **Like "Google Earth" for astronomers?**

PI: C. Aragon (NERSC)



Map of the sky as viewed from Palomar Observatory; color shows the number of times an area was observed

See Peter Nugent's NUG2009 Talk

- **Other NERSC gateways: GCRM (climate), Planck (Astro), Gauge Connection (QCD)**

NERSC 2009 Configuration

Large-Scale Computing System

Franklin (NERSC-5): Cray XT4
- 9,532 compute nodes; 38,128 cores
- ~25 Tflop/s on applications; 356 Tflop/s peak

Hopper (NERSC-6): Cray XT
- Phase 1: Cray XT5, 668 nodes, 5344 cores
- Phase 2: > 1 Pflop/s peak

Clusters

Jacquard and Bassi
- LNXI and IBM clusters
- Upgrading to Carver (NCS-c)

PDSF (HEP/NP)
- Linux cluster (~1K cores)

NERSC Global Filesystem (NGF)

Uses IBM's GPFS

440 TB; 5.5 GB/s

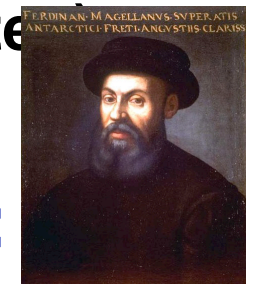
HPSS Archival Storage

- 59 PB capacity
- 11 Tape libraries
- 140 TB disk cache

Analytics / Visualization

Davinci (SGI Altix)
- Tesla testbed
- Upgrad planned

DOE Explores Cloud Computing

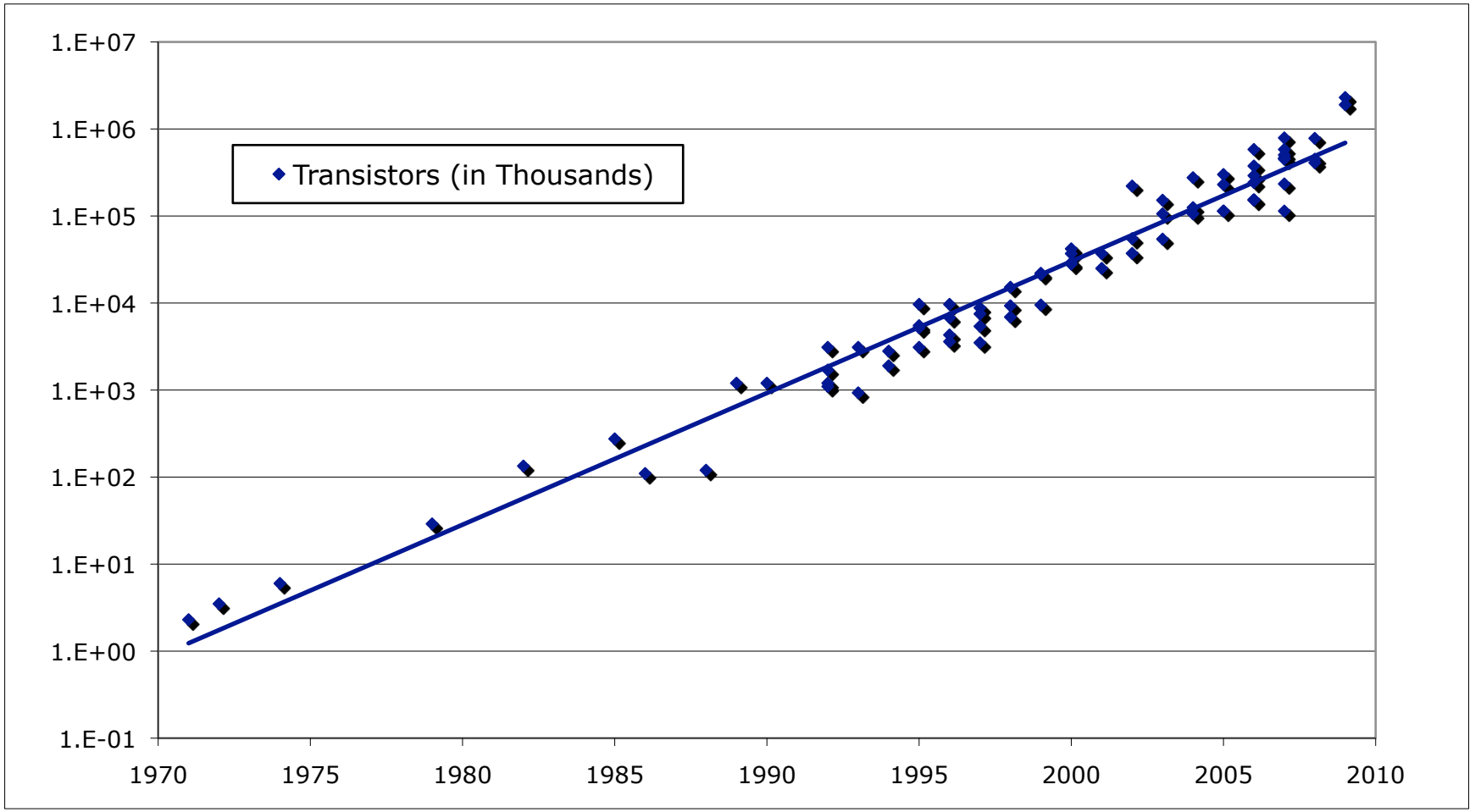


- **ASCR Magellan Project**
  - **$32M project at NERSC and ALCF**
  - **~100 TF/s compute cloud testbed (across site**
  - **Petabyte-scale storage cloud testbed**
- **Cloud questions to explore on Magellan:**
  - **Can a cloud serve DOE's mid-range computing needs?**

    → More efficient than cluster-per-PI model
  - **What part of the workload can be served on a cloud?**
  - **What features (hardware and software) are needed of a "Science Cloud"? (Eucalyptus at ALCF; Linux at NERSC)**
  - **How does this differ, if at all, from commercial**

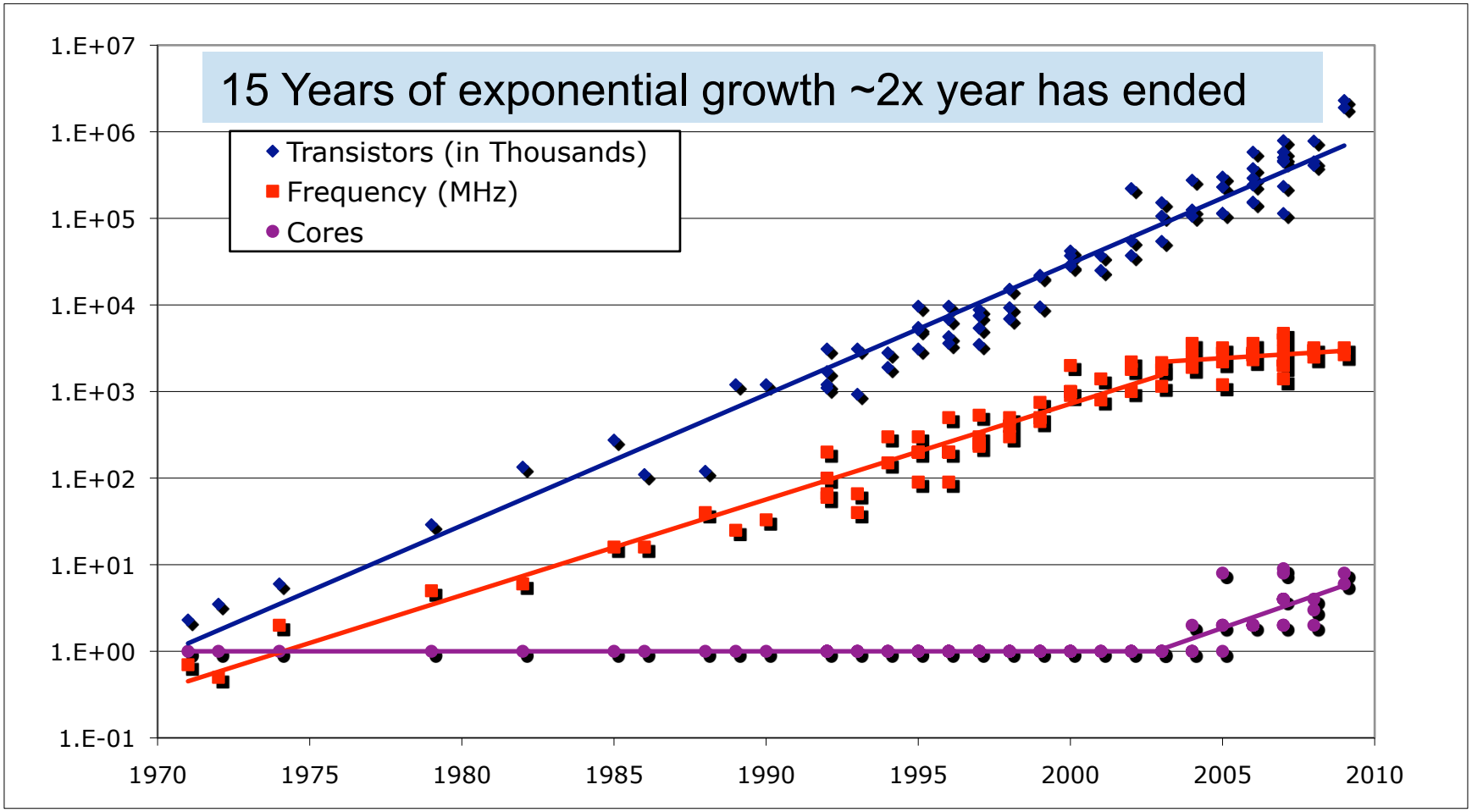
NATIONAL ENERGY RESEARCH SCIENTIFIC COMPUTING CENTER

Moore's Law is Alive and Well



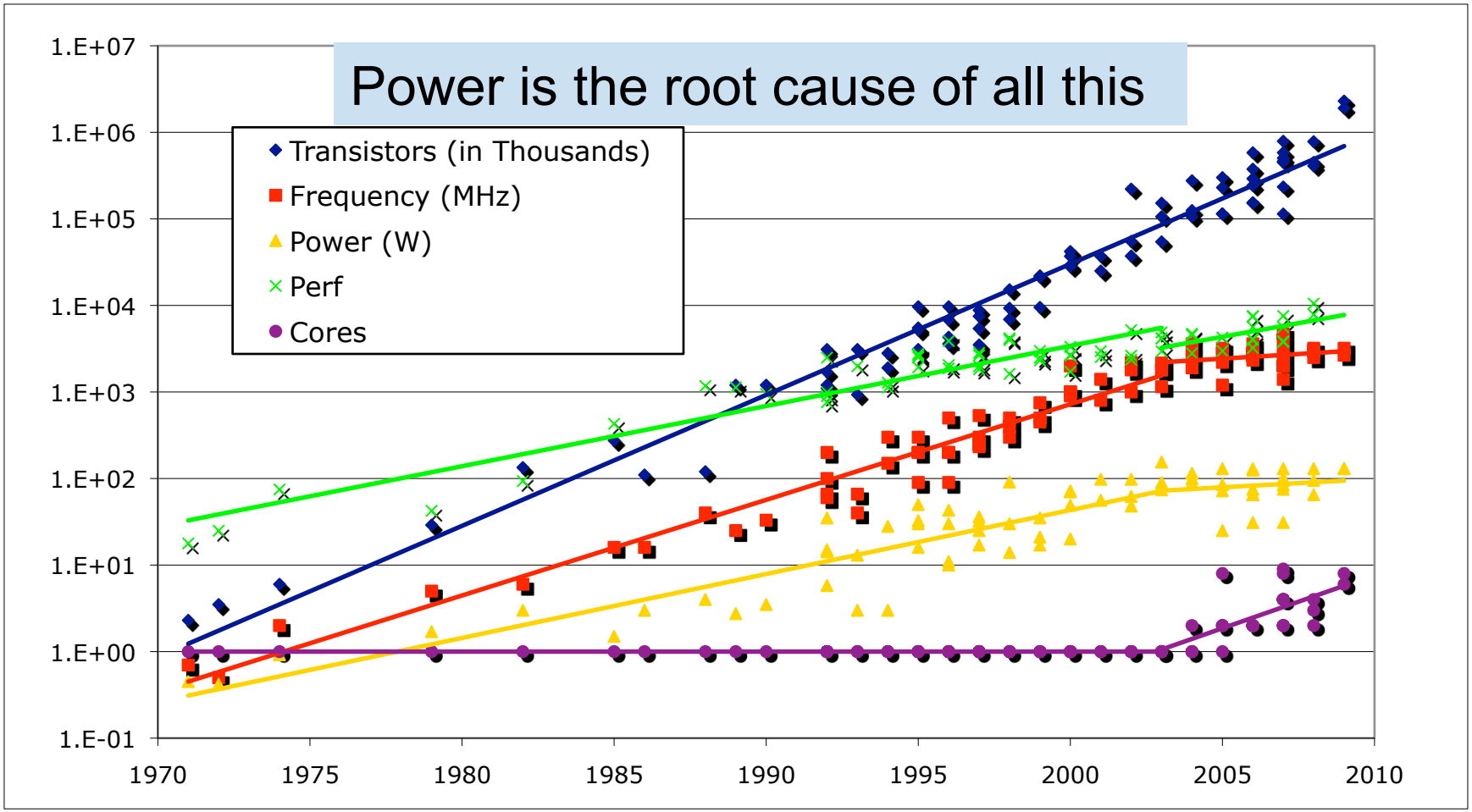
Data from Kunle Olukotun, Lance Hammond, Herb Sutter, Burton Smith, Chris Batten, and Krste Asanoviç

But Clock Frequency Scaling Replaced by Scaling Cores / Chip



Data from Kunle Olukotun, Lance Hammond, Herb Sutter, Burton Smith, Chris Batten, and Krste Asanoviç

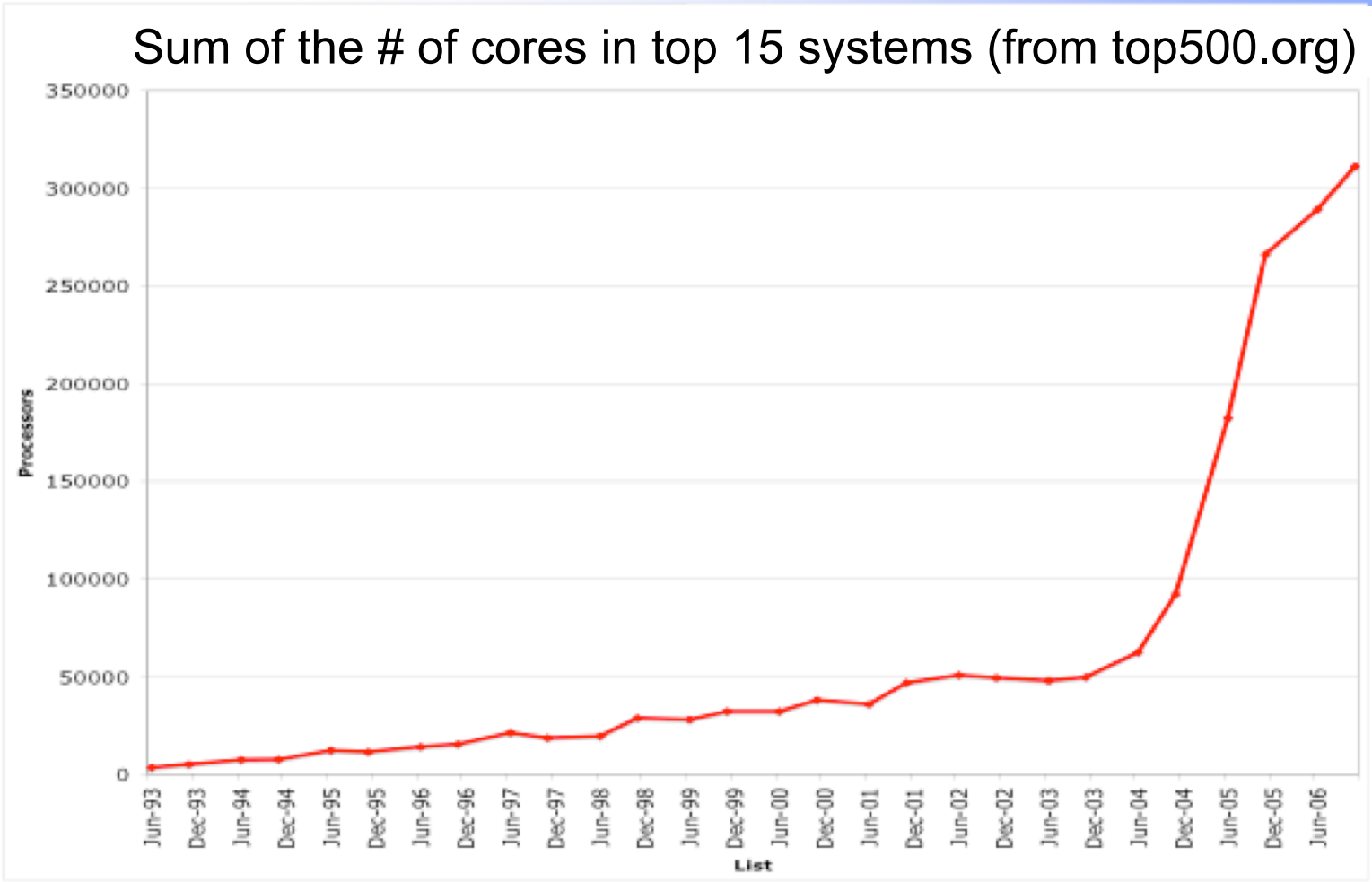
Performance Has Also Slowed, Along with Power



Data from Kunle Olukotun, Lance Hammond, Herb Sutter, Burton Smith, Chris Batten, and Krste Asanoviç

This has Also Impacted HPC System Concurrency

Sum of the # of cores in top 15 systems (from top500.org)

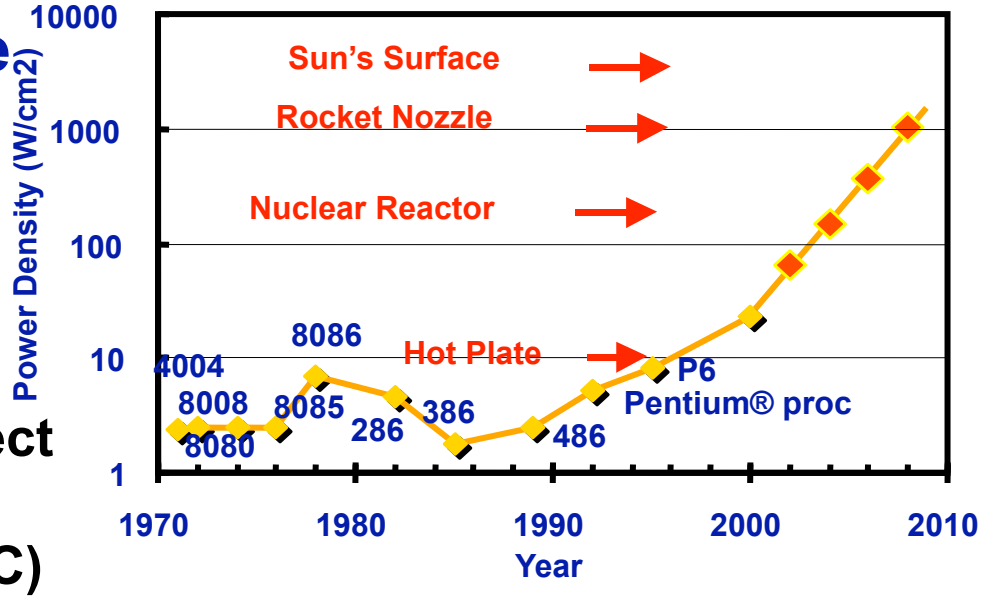


Exponential wave of increasing concurrency for forseeable future!

1M cores sooner than you think!

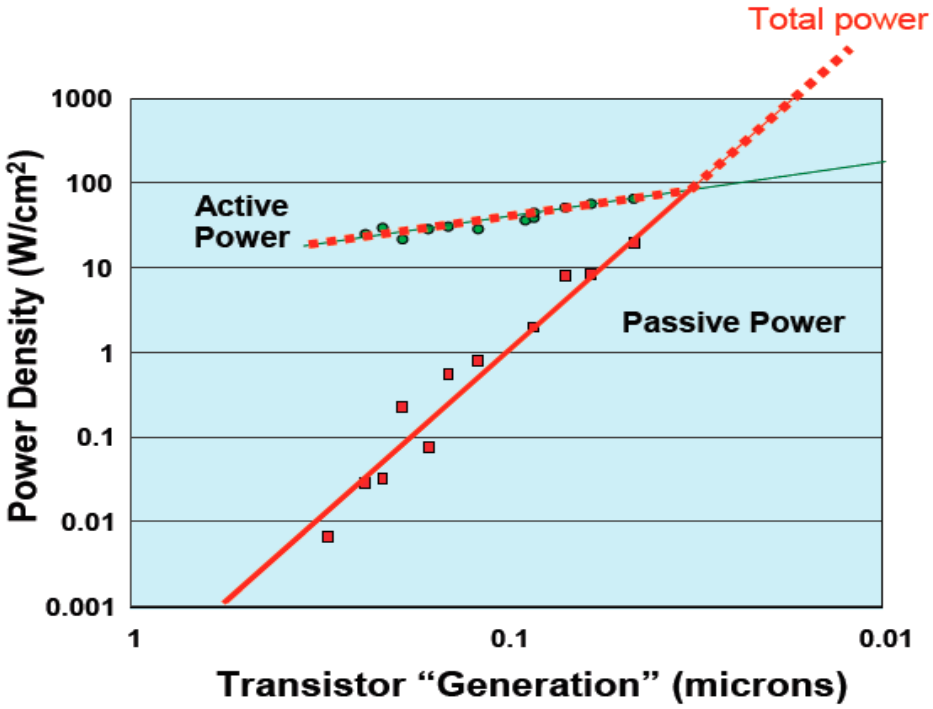
Parallelism is "Green"

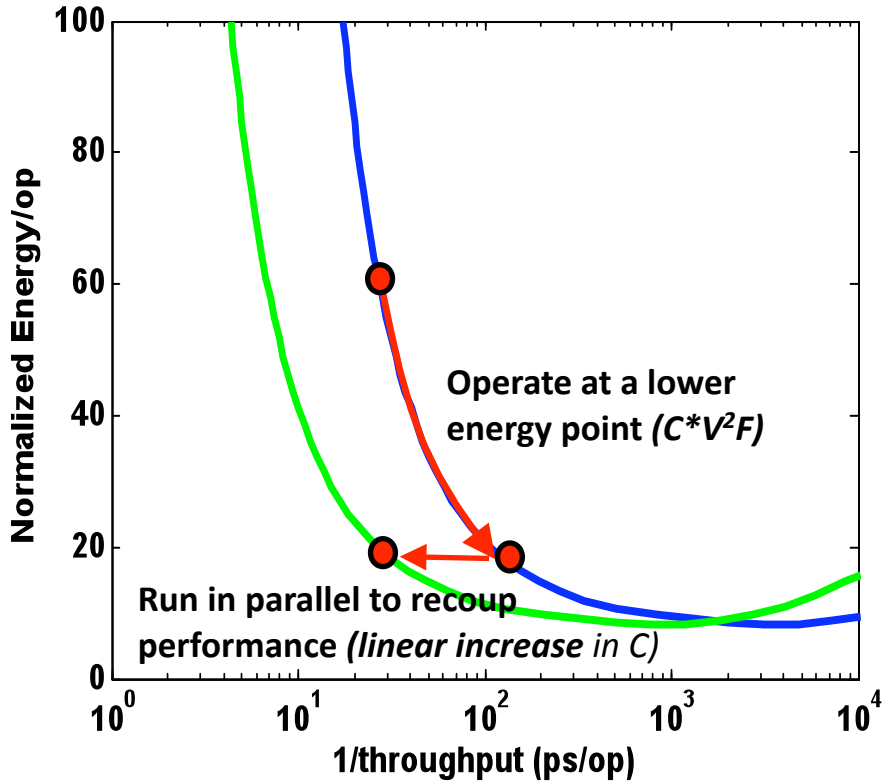
- **Concurrent systems are more power efficient**
  - Dynamic power is proportional to V^2fC
  - Increasing frequency (f) also increases supply voltage (V) → cubic effect
  - Increasing cores increases capacitance (C) but only linearly



- **High performance serial processors waste power**
  - Speculation, dynamic dependence checking, etc. burn power
  - Implicit parallelism discovery
- **Question: *Can you double the concurrency in your algorithms and software every 2 years?***

Parallelism to Recover Performance



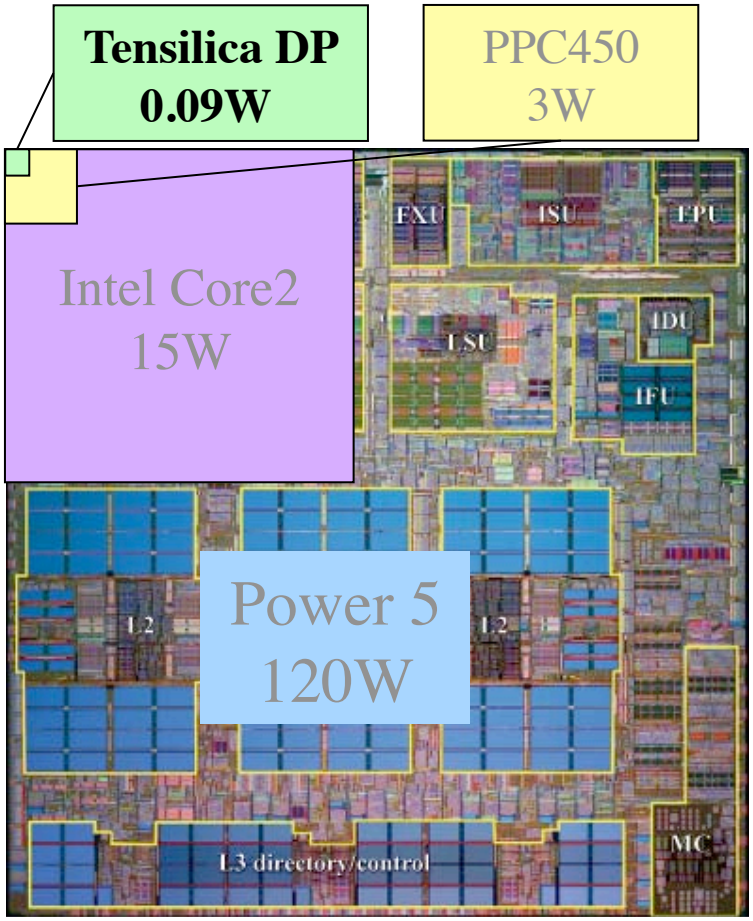


- **Computing performance is now limited by power dissipation. This has forced the move to parallelism as principal means of increasing performance without increasing energy per operation.**

Traditional Sources of Performance Improvement are Flat-Lining

- **15 years of *exponential* clock speed growth has ended**
- **Hardware to automatically extract parallelism has little promise for future increases:**
  - **Instruction Level Parallelism has been tapped out**
  - **Speculative execution wastes power for small speedup**
  - **Chips were over-engineering to run purely serial code**
- **How to use the transistors?**
  - Industry "Conservative" Response is ***Multicore***: #cores per chip doubles every 18 months *instead* of clock frequency!
  - Pressure for smaller simpler "cores" sometimes data parallelism:
    - Simpler cores are more power efficient than serial-optimized cores
    - Data parallelism (SIMD, vectors, streaming processors, GPUs) avoid control overhead of full core per functional unit

The Case for Small Simple Cores



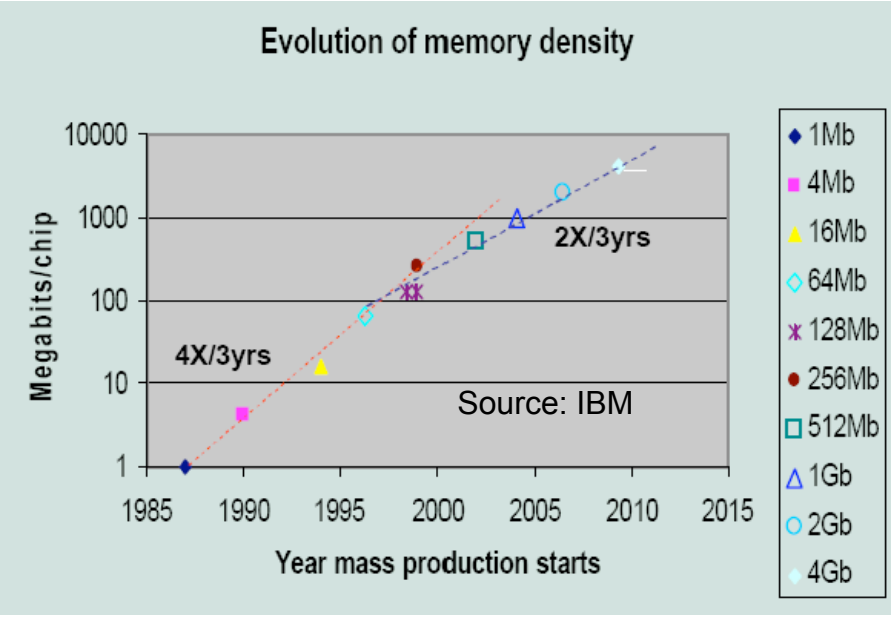
- **IBM Power5 (server)**
  - **120W@1900MHz**
  - **Baseline**
- **Intel Core2 sc (laptop) :**
  - **15W@1000MHz**
  - ***4x more FLOPs/watt than baseline***
- **IBM PPC 450 (BG/P - low power)**
  - **0.625W@800MHz**
  - **90x more**
- **Tensilica XTensa (Moto Razor + DP) :**
  - **0.09W@600MHz**
  - **400x more**

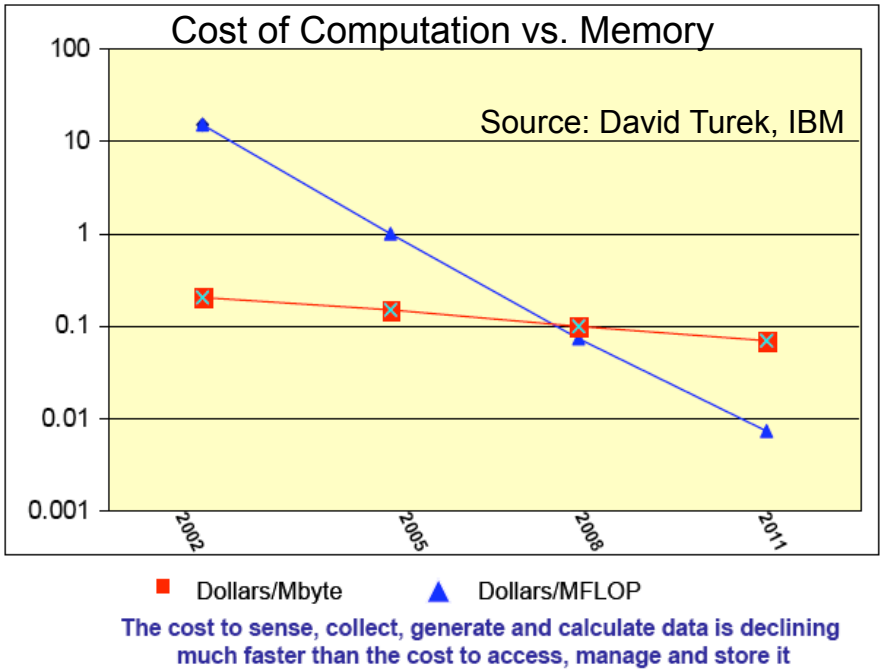
1/3 the efficiency per core, but 1/400th of the power

Memory is Not Keeping Pace

Technology trends against a constant or increasing memory per core

- Memory density is doubling every three years; processor logic is every two
- Storage costs (dollars/Mbyte) are dropping gradually compared to logic costs





Question: *Can you double concurrency without doubling memory?*

NATIONAL ENERGY RESEARCH SCIENTIFIC COMPUTING CENTER

The *NEW* Scaling Rules

- **Old Trend**
  - Clock frequency doubles every 18 months
  - Terascale to Petascale was done without changing Programming models (Giga to Tera was not so easy)
  - One order of magnitude was clock speed
    - ASCI Red originally had 200 Mflop/s processors
- **New Trend**
  - Number of cores per chip will at least double every two years
  - Clock speed will not increase (possibly decrease)
  - Very simple "cores" with data parallelism will be popular: wider SIMD, GPUs, accelerators
- **Conclusion: Exascale Need to deal with systems with billion-way concurrency**
  - ***Some will be fine-grained parallelism (SIMD, GPUs,…)***

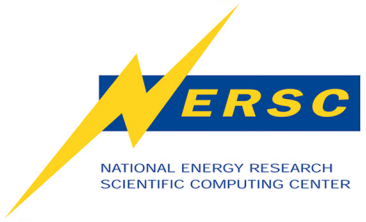
    ~10^9 cycles/s/core * 10^9 concurrency = 10^{18}

Future of Memory Scaling

- **Old Trend**
  - Memory increased proportional to CPU performance (more memory per core)
  - Scale-up problem proportional to system parallelism (weak scaling)
- **New Trend**
  - Memory per core will *decrease* (slow increase per node)
  - Strong-scaling (increase parallelism with fixed problem size) will important to replace clock speed scaling
- **Conclusion: Strong and weak scaling will be important**
  - Need strong-scaling to keep runtimes from growing exponentially with increased problem size (clock no faster)
  - Less memory per core (*strong-scaling*)
  - Weak scaling will be used

Software Issues at Scale

- **Power concerns will dominates all others;**
  - Concurrency is the most significant knob we have: lower clock, increase parallelism
  - Power density and facility energy
- **Summary Issues for Software**
  - 1EF system: Billion-way concurrency, O(1K) cores per chip
  - 1 PF system: millions of threads and O(1K) cores per chip
  - The memory capacity/core ratio may drop significantly
  - Faults will become more prevalent
  - Flops are cheap relative to data movement
  - "Core" is a retro term: "sea of functional units"
    - 1 thread per functional unit
    - Many functional units per thread (SIMD, vectors)
    - Many threads per functional unit (Niagra)

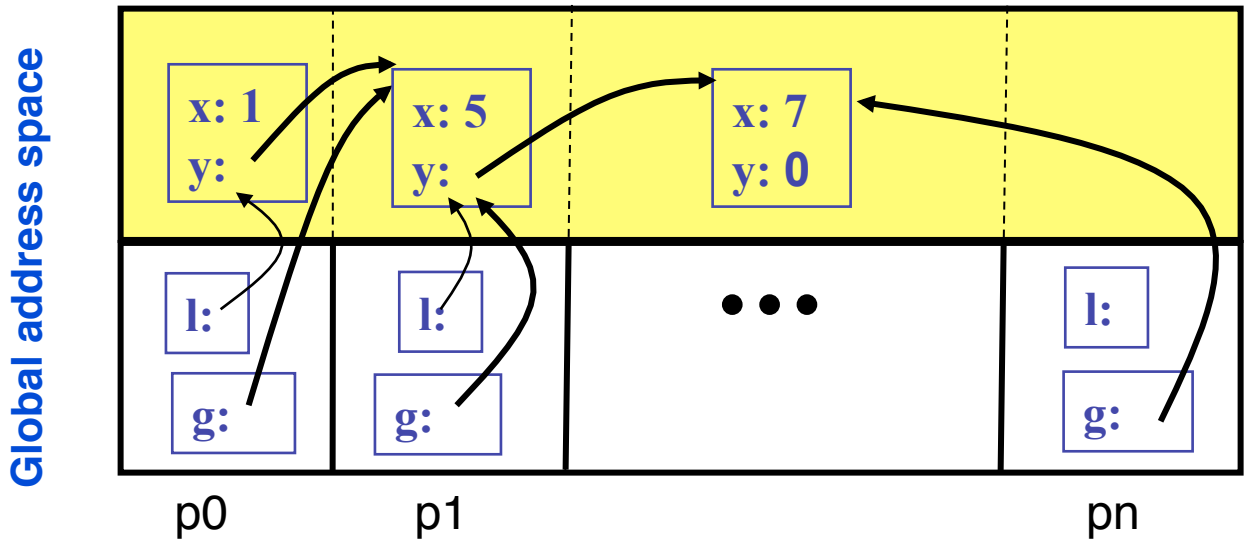


If the Answer is MPI+X: What is X?

- Multicore needs a programming model, whether it is inside a supercomputer/cluster or on it's own
- MPI will not disappear as the X HPC programming model
- But wee need something, X, for multicore
- X is probably not OpenMP
  - Too much serial thinking leads to over-synchronization
  - Poor expression of locality (will not scale)
- X might be UPC or PGAS language
  - Explicit definition of local vs. remote
  - Very lightweight communication
- X might be CUDA or OpenCL
  - OpenCL is very CUDA-like cross-platform extension to C language
  - CUDA is also being extended to also taret multicore

PGAS Languages: Why use 2 Programming Models when 1 will do?

- *Global address space:* thread may directly read/write remote data
- *Partitioned:* data is designated as local or global



- Remote put and get: never have to say "receive"
  - Remote function invocation? See HPCS languages
- No less scalable than MPI!
- Permits sharing, whereas MPI rules it out!
- One model rather than two, but if you insist on two:
  - Can call UPC from MPI and vice verse (tested and used)

NERSC
NATIONAL ENERGY RESEARCH SCIENTIFIC COMPUTING CENTER

Things Software *Should* Do

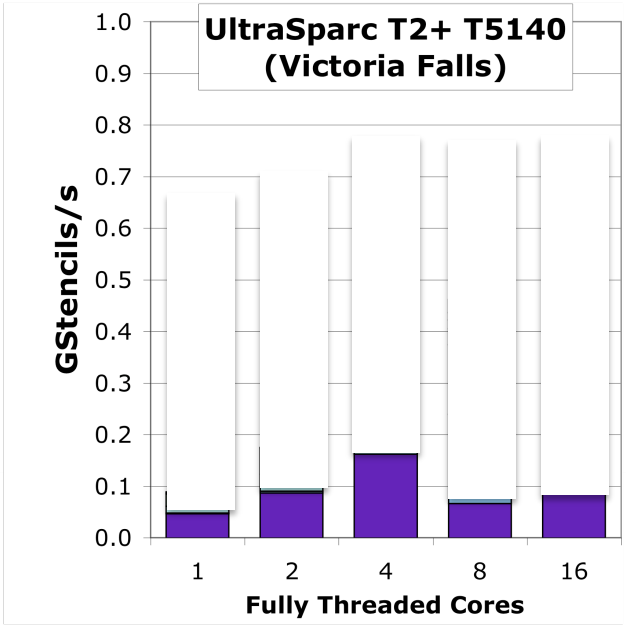
(And some encouraging evidence that it can)

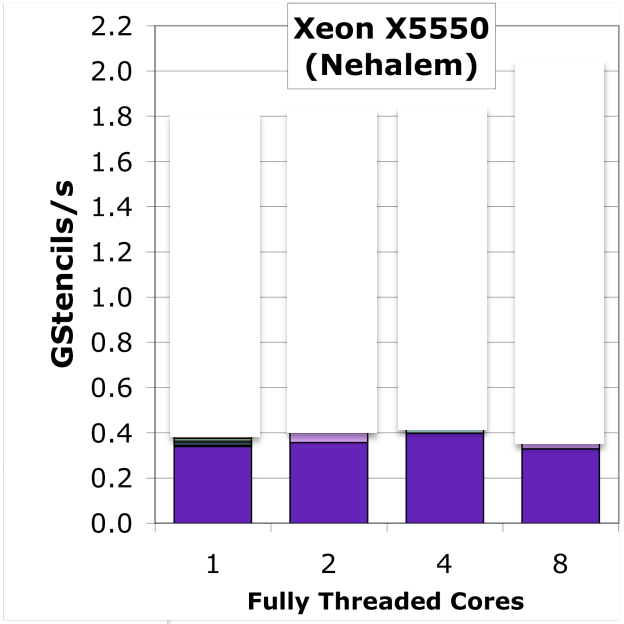
#1) Software Needs to Avoid Unnecessary Bandwidth Use

Nearest-neighbor 7point stencil on a 3D array

Use Autotuning!

Write code generators and let computers do tuning



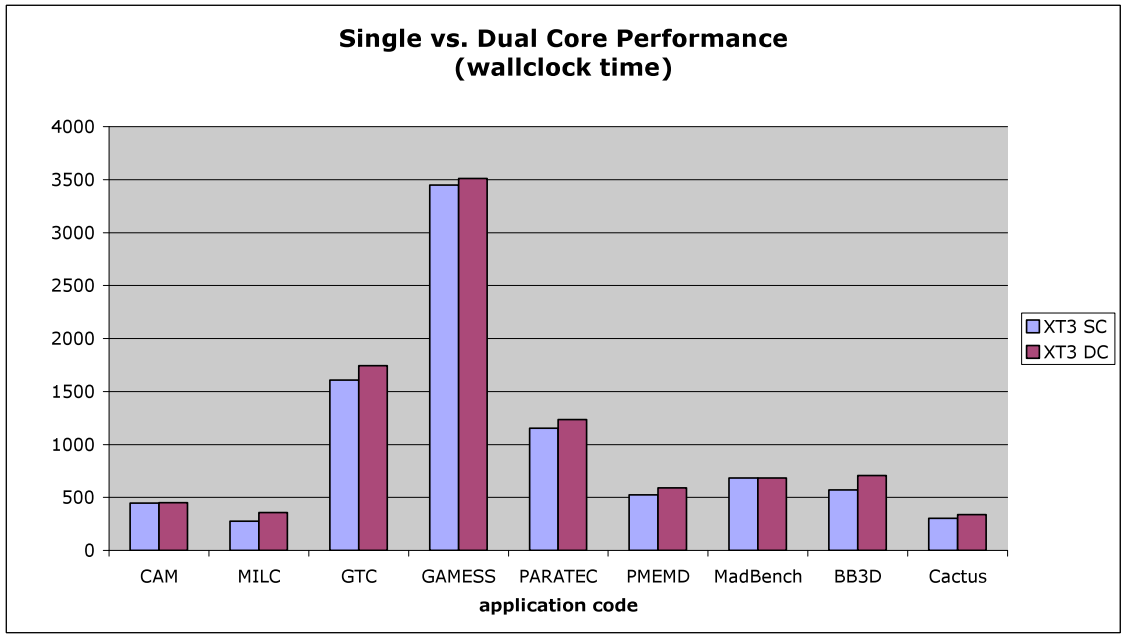


Reference (cache) Implementation

#2) Software Needs to Address Little's Law (waiting on Latency)

Little's Law: required concurrency = bandwidth * latency

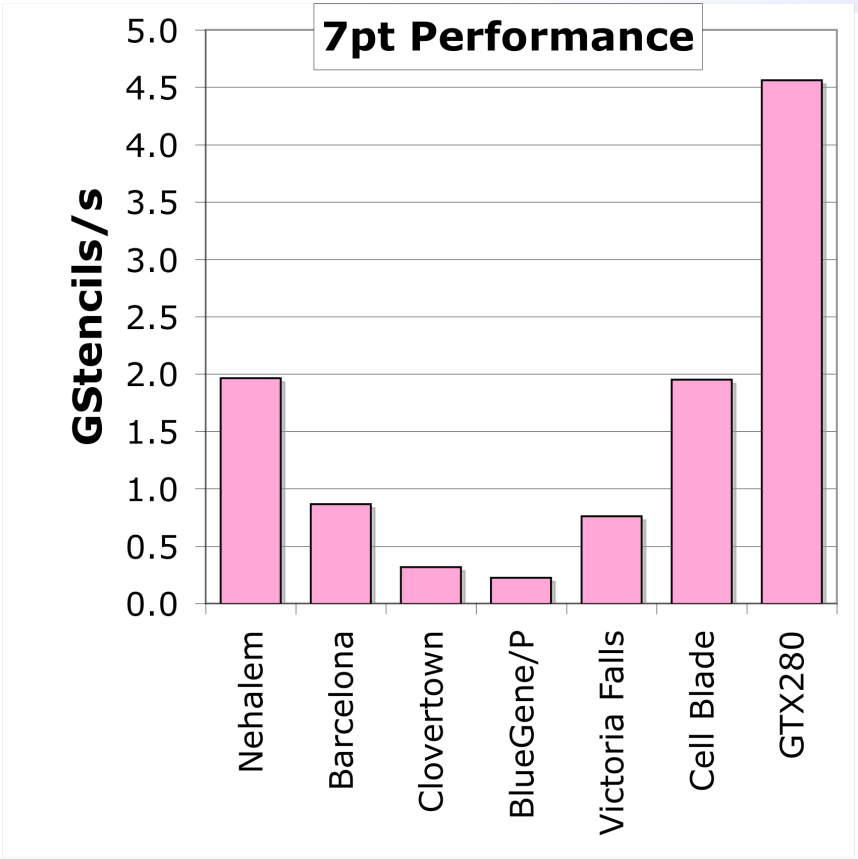
#outstanding_memory_fetches = bandwidth* latency

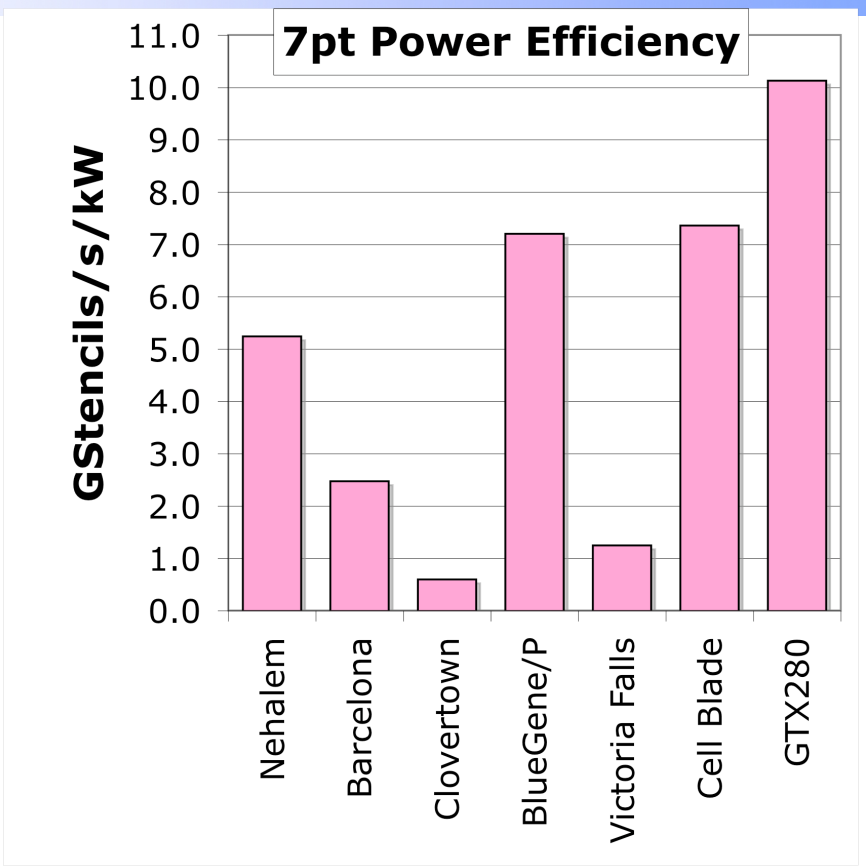


NERSC application benchmarks
Shalf et al

Experiment: Running on a fixed number of cores
1 core per socket vs 2 cores per socket
Only 10% performance drop from sharing (halving) bandwidth

7 Point Stencil Revisited





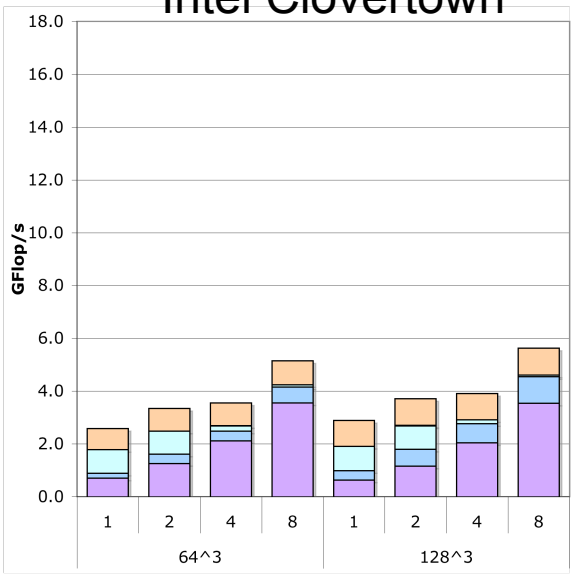
- Cell and GTX280 are notable for both performance and energy efficiency due to their explicitly manage memory

Joint work with Kaushik Datta, Jonathan Carter, Shoaib Kamil, Lenny Oliker, John Shalf, and Sam Williams

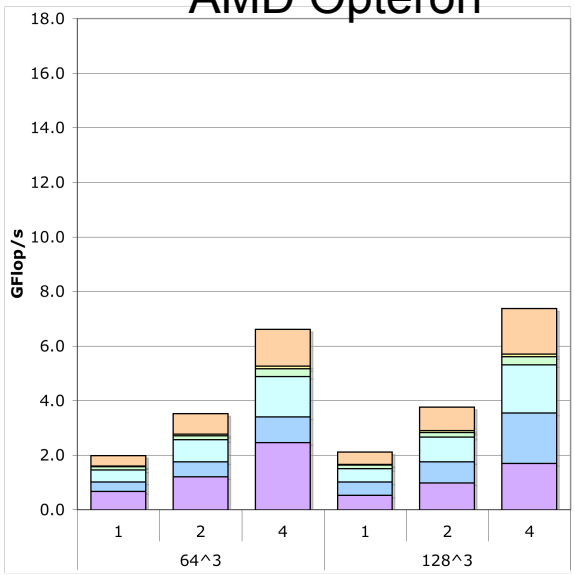
#3) Use Novel Hardware Features Through Code Generators

NATIONAL ENERGY RESEARCH SCIENTIFIC COMPUTING CENTER

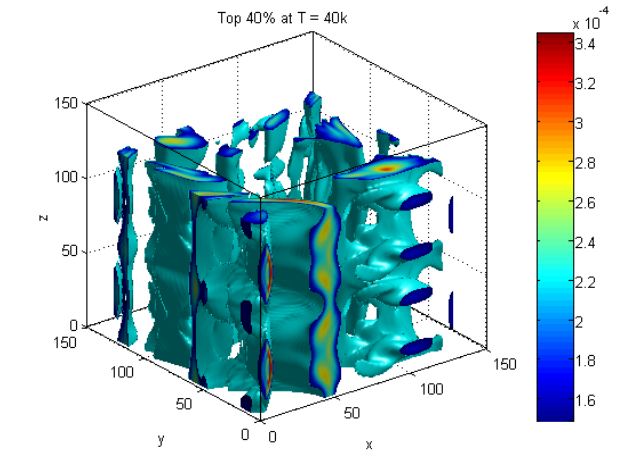
Intel Clovertown



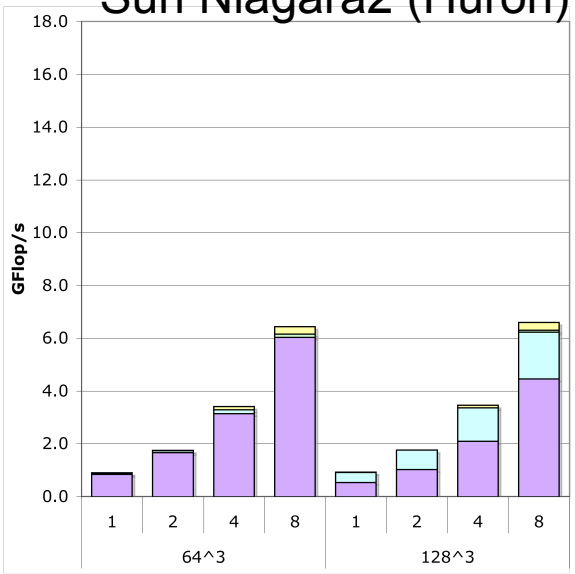
AMD Opteron



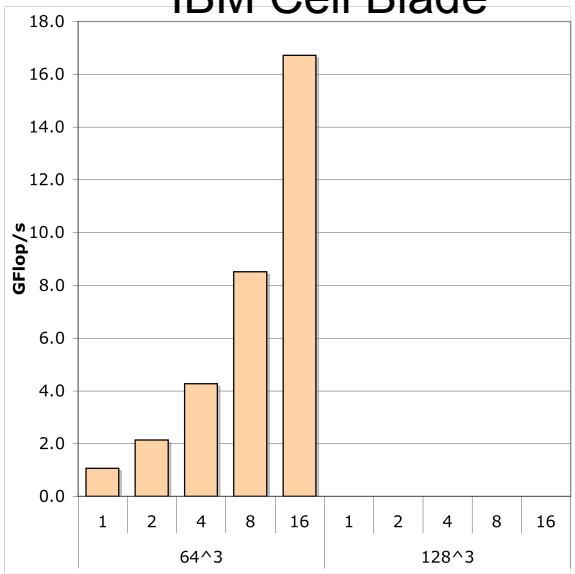
LBMHD is not always bandwidth limited: used SIMD, etc.



Sun Niagara2 (Huron)



IBM Cell Blade*



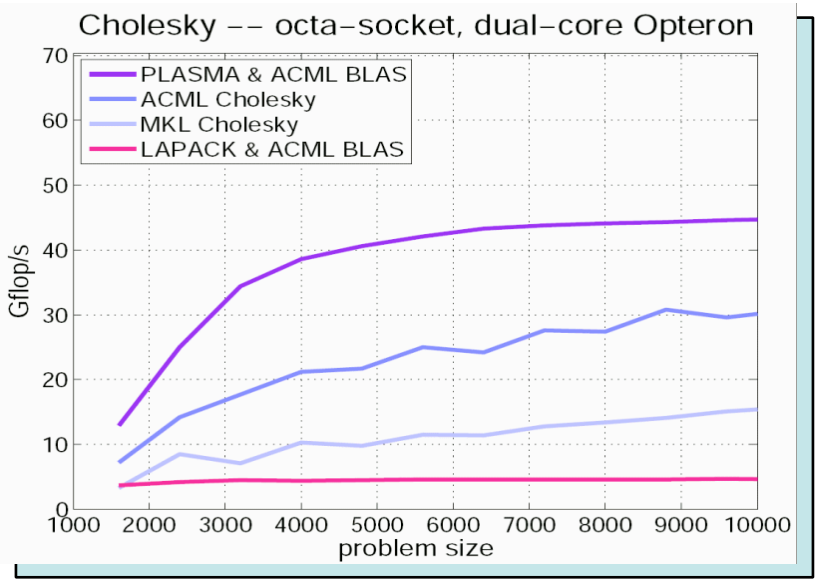
- +SIMDization
- +SW Prefetching
- +Unrolling
- +Vectorization
- +Padding
- Naïve+NUMA

Joint work with Sam Williams, Lenny Oliker, John Shalf, and Jonathan Carter

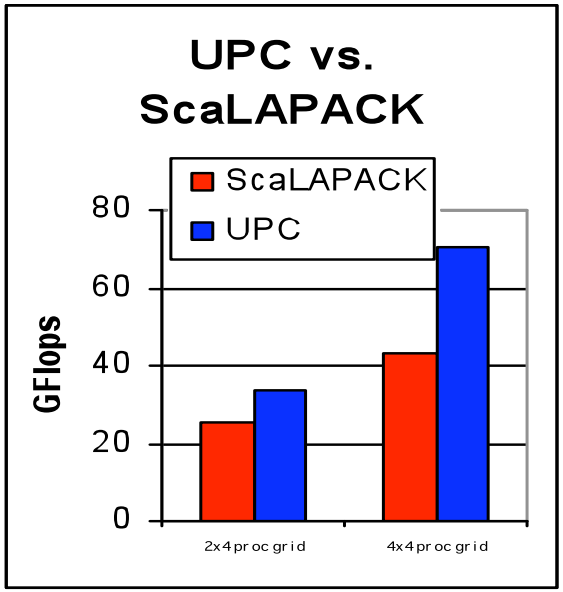
BERKELEY LAB

#4) Avoid Unnecessary Global Synchronization

PLASMA on shared memory



UPC on partitioned memory



UPC LU factorization code adds cooperative (non-preemptive) threads for latency hiding

- **New problem in partitioned memory: allocator deadlock**
- **Can run on of memory locally due tounlucky execution order**

PLASMA by Dongarra et al; UPC LU joint with Parray Husbands

#5) Avoid Unnecessary Synchronization in Point-to-Point Communication

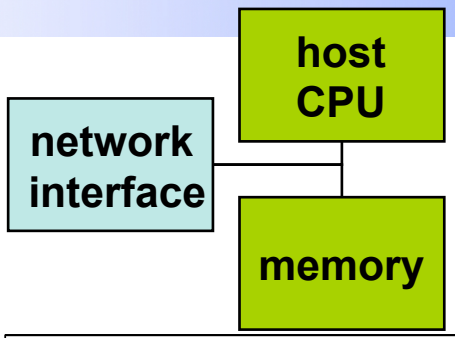
NATIONAL ENERGY RESEARCH SCIENTIFIC COMPUTING CENTER

two-sided message

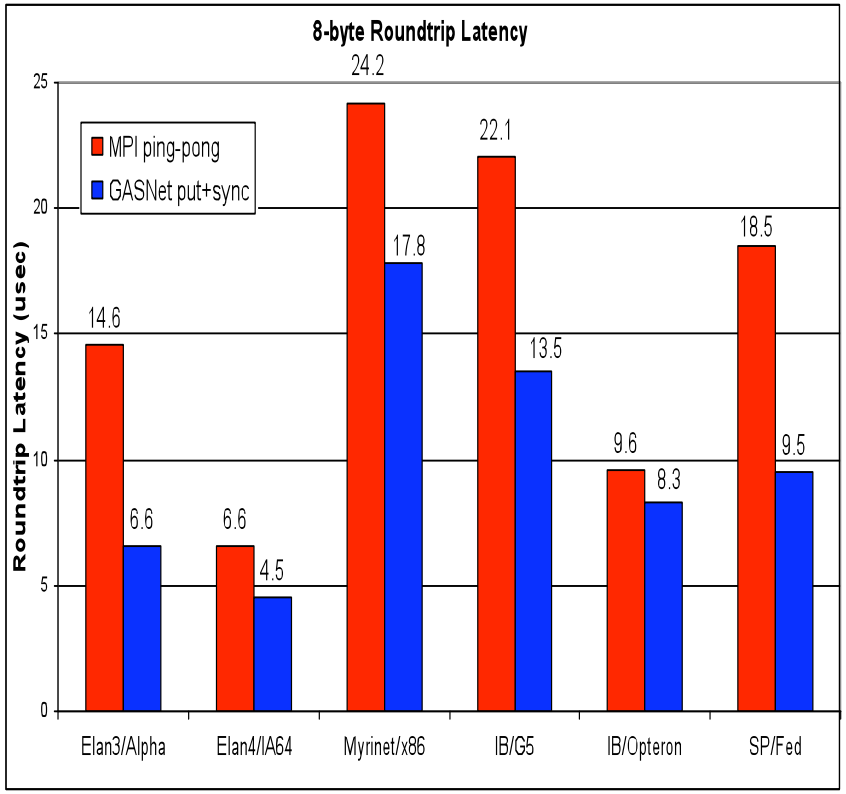
message id	data payload

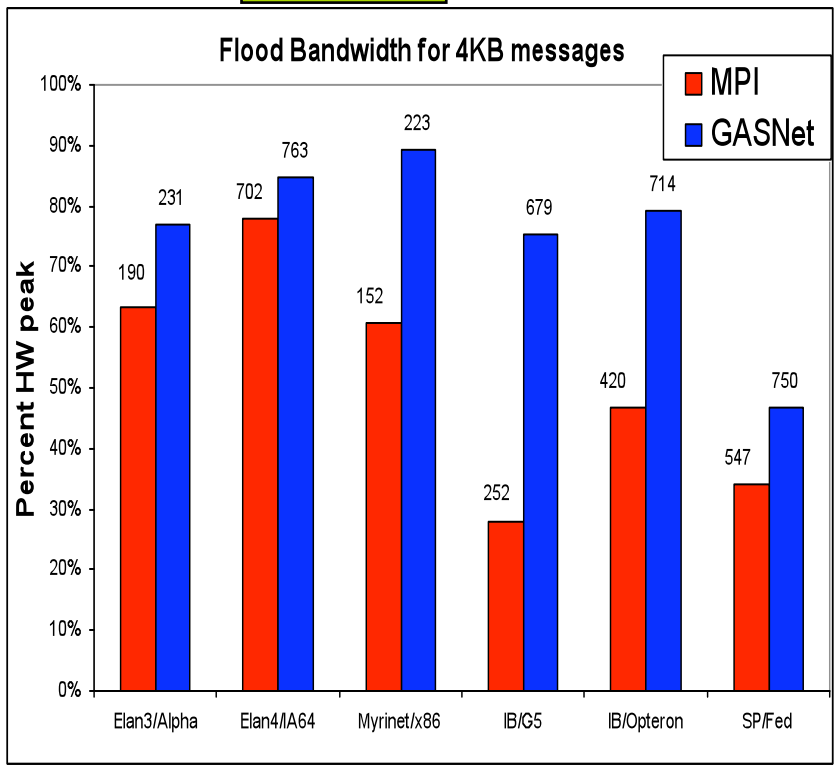
one-sided put message

address	data payload



Pay only for what you need





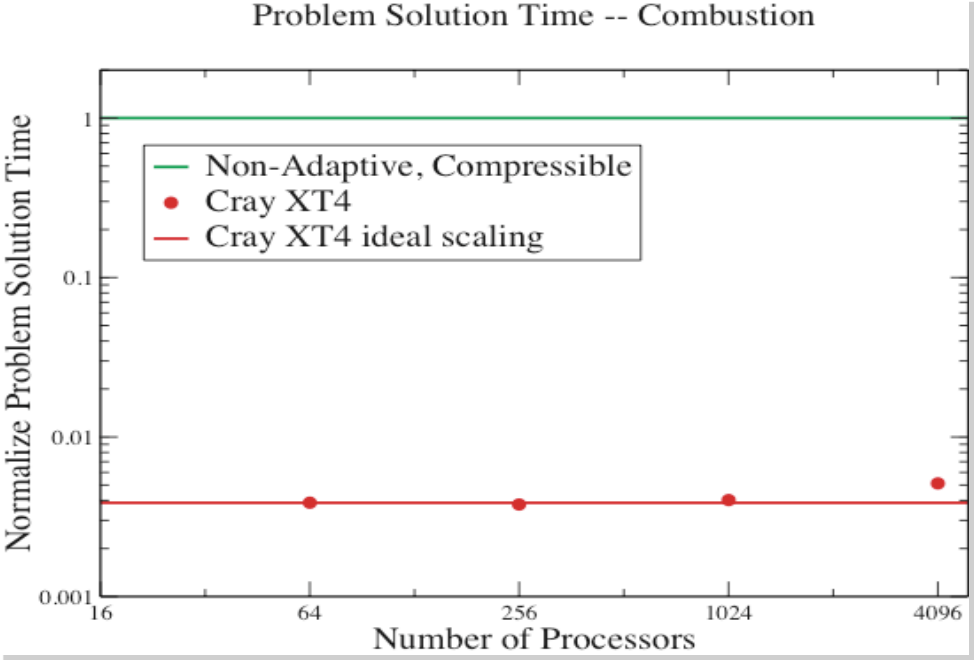
U.S. DEPARTMENT OF ENERGY | Office of Science

Joint work with Berkeley UPC Group

BERKELEY LAB

#6) Make use of Good (Algorithmically Scalable) Algorithms

- **Algorithmic gains in last decade have far outstripped Moore's Law**
  - **Adaptive meshes rather than uniform**
  - **Sparse matrices rather than dense**
  - **Reformulation of problem back to basics**

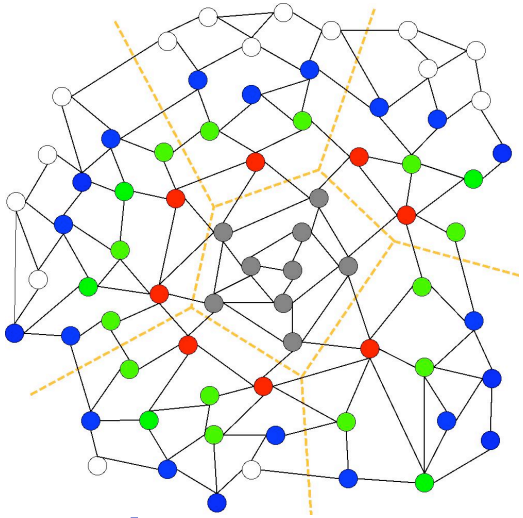


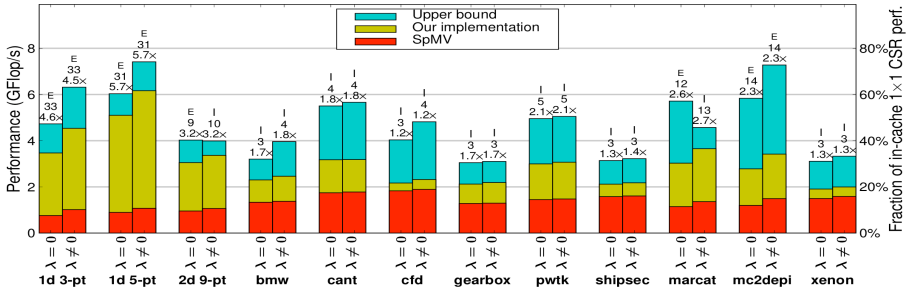
- **Example of canonical "Poisson" problem on n points:**
  - **Dense LU: most general, but $O(n^3)$ flops on $O(n^2)$ data**
  - **Multigrid: fastest/smallest, O(n) flops on O(n) data**

Performance results: John Bell et al

#7) Algorithm Developers should Avoid Communication, not Flops

- Consider Sparse Iterative Methods
  - Nearest neighbor communication on a mesh
  - Dominated by time to read matrix (edges) from DRAM
  - And (small) communication and global synchronization events at each step
    - **Can we lower data movement costs?**
- Take *k* steps "at once" with one matrix read from DRAM and one communication phase
  - **Parallel implementation**
  - O(log p) messages vs. O(k log p)
  - **Serial implementation**
  - O(1) moves of data moves vs. O(k)
- Performance of $A^k x$ operation relative to Ax and upper boun
  - **Runs up to 5x faster on SMP**

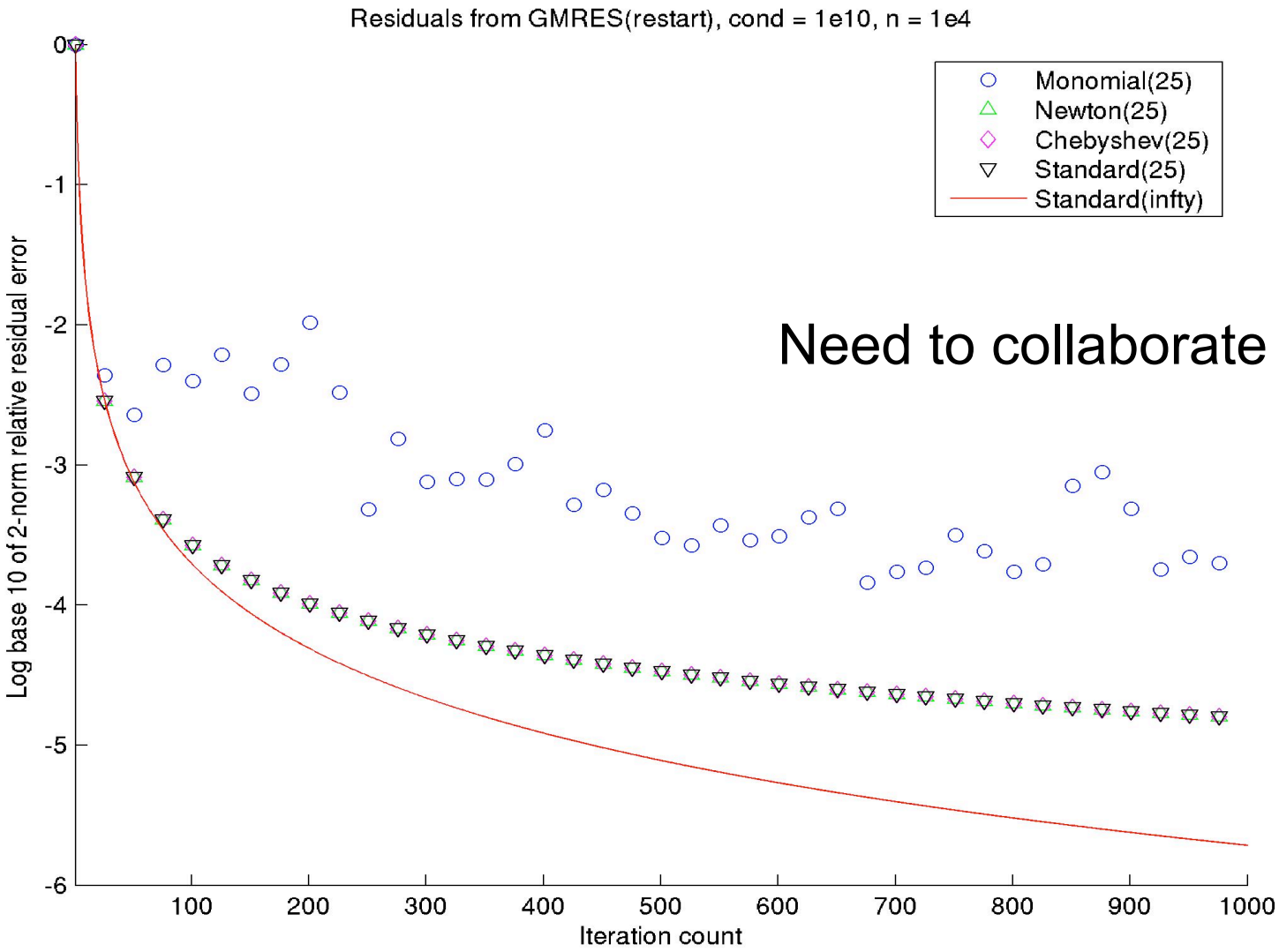




Joint work with Jim Demmel, Mark Hoemman, Marghoob Mohiyuddin

U.S. DEPARTMENT OF ENERGY | Office of Science

But the Numerics have to Change!



Need to collaborate

Work by Jim Demmel and Mark Hoemman

Multicore Rules for Software (and Algorithms and Applications)

1) Don’t waste memory bandwidth
2) Remember Little's Law
3) Use novel hardware features
4) Avoid global synchronization
5) Avoid point-to-point synchronization (clusters)
6) Choose efficient algorithms
7) Rethink algorithms to avoid data movement

NERSC
NATIONAL ENERGY RESEARCH SCIENTIFIC COMPUTING CENTER

Conclusions

- Single processors will not get faster
- Memory per core will likely drop
- Strong scaling (parallelism) will be important for all applications
  - Need parallel software model
  - OpeMP, UPC/CAF, or CUDA/OpenCL
- Power and energy costs will dominate
- Work with experts on software, algorithms, applications

NERSC
NATIONAL ENERGY RESEARCH SCIENTIFIC COMPUTING CENTER

More Info

- The Berkeley View/Parlab
  - http://view.eecs.berkeley.edu
  - http://parlab.eecs.berkeley.edu/
- Berkeley Autotuning and PGAS projects
  - http://bebop.cs.berkeley.edu
  - http://upc.lbl.gov
  - http://titanium.cs.berkeley.edu
- NERSC System Architecture Group
  - http://www.nersc.gov/projects/SDSA
- LBNL Future Technologies Group

  http://crd.lbl.gov/ftg